

MOSTEK®

MD SERIES MICROCOMPUTER MODULES

Operations Manual

**MDX-SC/D
SYSTEM CONTROLLER
DIAGNOSTIC MODULE**

MOSTEK

MD SERIES MICROCOMPUTER MODULES

OPERATIONS MANUAL

SYSTEM CONTROLLER/DIAGNOSTIC MODULE

MDX - SC/D

OPERATION MANUAL

FOR

MDX - SC/D

(MK77963)

TABLE OF CONTENTS

SECTION NUMBER	PARAGRAPH NUMBER	TITLE	PAGE NUMBER
1		GENERAL INFORMATION	
	1-1	INTRODUCTION	1-1
	1-4	MD SERIES GENERAL DESCRIPTION	1-1
	1-7	FEATURES	1-2
2		HARDWARE FUNCTIONAL DESCRIPTION	
	2-1	INTRODUCTION	2-1
	2-3	MEMORY ARRAY	2-1
	2-4	MEMORY SELECTION LOGIC	2-1
	2-5	PORT SELECTION LOGIC	2-1
	2-6	OPERATOR INTERFACE	2-1
	2-7	INTERRUPTS	2-2
3		USER-SELECTABLE OPTIONS	
	3-1	EPROM DECODING JUMPER OPTIONS	3-1
	3-3	ADDRESS CONFIGURATION STRAPPING	3-1
	3-5	RESET/START SWITCH STRAPPING	3-1
4		SOFTWARE FUNCTIONAL DESCRIPTION	
	4-1	INTRODUCTION	4-1
	4-3	PIO PROGRAMMING	4-1
	4-4	INTRODUCTION	4-1
	4-6	I/O ADDRESSING	4-1
	4-8	PIO OPERATION	4-2
	4-10	PIO PORT A PROGRAMMING	4-2
	4-21	PIO PORT B PROGRAMMING	4-5
	4-25	PIO PORT A DATA CONFIGURATION	4-6

TABLE OF CONTENTS (contd)

SECTION NUMBER	PARAGRAPH NUMBER	TITLE	PAGE NUMBER
4 cont.	4-27	PIO PORT B DATA CONFIGURATION	4-7
	4-29	READING THE THUMBWHEELS	4-8
	4-31	USING THE DISPLAY	4-9
	4-33	READING THE RESET/START SWITCH	4-9
	4-35	SWITCH STRAPPING	4-10
	4-36	INTRODUCTION	4-10
	4-38	CONFIGURATIONS	4-10
	4-41	MEDEX-80	4-15
	4-42	INTRODUCTION	4-15
	4-46	THEORY OF OPERATION	4-17
	4-48	CONCEPT OF DIAGNOSTIC MODULE	4-18
	4-49	CONCEPT OF TEST DIRECTIVE TABLE	4-18
	4-60	CONCEPT OF DIAGNOSTIC INTERRUPT VECTOR TABLE	4-21
	4-67	INTERRUPTS	4-24
	4-71	DIAGNOSTIC SYSTEM ARCHITECTURE	4-28
	4-75	MEDEX-80 DISKETTE	4-28
	4-82	SUMMARY	4-30
5		SPECIFICATIONS	
	5-1	INTRODUCTION	5-1
	5-3	ELECTRICAL SPECIFICATIONS	5-1
	5-13	MECHANICAL SPECIFICATIONS	5-2
APPENDIX	A	FACTORY REPAIR LIMITED WARRANTY	A-1 A-1
APPENDIX	B	SCHEMATIC	B-1

TABLE OF CONTENTS (contd)

SECTION NUMBER	PARAGRAPH NUMBER	TITLE	PAGE NUMBER
APPENDIX	C	ASSEMBLY DRAWING	C-1
		PARTS LIST	C-2
APPENDIX	D	MEDEX-80 OPERATOR DISPLAY CHARACTERS	D-1

LIST OF FIGURES

FIGURE NUMBER	TITLE	PAGE NUMBER
2-1	BOARD PHOTO WITH OVERLAY	2-3
2-2	MDX-SC/D BLOCK DIAGRAM	2-4
4-1	MEDEX-80 SYSTEM	4-16
4-2	TEST DIRECTIVE TABLE SOURCE CODE EXAMPLE	4-22
4-3	DIAGNOSTIC MODULE RELATIONSHIP TO TEST DIRECTIVE TABLE	4-23
4-4	DIAGNOSTIC INTERRUPT VECTOR TABLE SOURCE CODE EXAMPLE	4-26
4-5	DIAGNOSTIC MODULE RELATIONSHIP TO DIAGNOSTIC INTERRUPT VECTOR TABLE	4-27
4-6	MEDEX-80 SYSTEM ARCHITECTURE	4-29

LIST OF TABLES

TABLE NUMBER	TITLE	PAGE NUMBER
3-1	EPROM STRAPPING OPTIONS	3-2
4-1	RESET/START SWITCH USER STRAP OPTIONS	4-12
4-2	PORT CONTROL LOADING REQUIREMENTS FOR SWITCH CONFIGURATIONS	4-14

SECTION 1

GENERAL INFORMATION

1-1. INTRODUCTION

1-2. The MDX-SC/D was designed as a system controller/diagnostic card. As a system controller card, it is equipped with a jumper option to provide the user with a Reset/Interrupt line to the CPU. This signal can be utilized as a hardware system Reset.

1-3. As a diagnostics card, it is equipped with thumbwheel switches for test control, a Reset/Start switch for beginning the test, and two LED displays for observing the results of the test.

1-4. MD SERIES GENERAL DESCRIPTION

1-5. The MD series and the STD BUS were designed to satisfy the need for low-cost OEM microcomputer modules. The STD BUS uses a mother board interconnect system concept and is designed to handle any MD series card type in any slot. The modules for the STD BUS are a compact 4.5 x 6.5 inches, providing for system partitioning by function (RAM, EPROM, I/O). The smaller module size makes system packaging easier while increasing MOS-LSI densities provide high functionality per module.

1-6. The MD series of OEM microcomputer boards and the STD BUS offer the most cost-effective system configuration available to the OEM system designer.

1-7. FEATURES

- Provides an operator interface with switches/displays.
- 10K x 8 EPROM (2716's not included).
- Dual purpose card; System Controller and/or Diagnostic I/F.
- Interrupt driven programmability
- Address strap selectable.
- 4 MHz option.

SECTION 2

HARDWARE FUNCTIONAL DESCRIPTION

2-1. INTRODUCTION

2-2. This section describes the hardware operation of the MDX-SC/D card. Figure 2-1 shows the board photo with overlays and Figure 2-2 shows the block diagram of the card.

2-3. **MEMORY ARRAY.** The memory array consists of up to 5 MK2716's (2K x 8, +5V only EPROM). The total storage of the MDX-SC/D board is 10,240 bytes. The EPROM's can contain the diagnostic program necessary for testing the modules.

2-4. **MEMORY SELECTION LOGIC.** The memory selection logic is responsible for decoding the selected address and enabling the selected EPROM on to the data bus through the data bus buffer.

2-5. **PORT SELECTION LOGIC.** The port selection logic is used to allow read and write operations to the PIO.

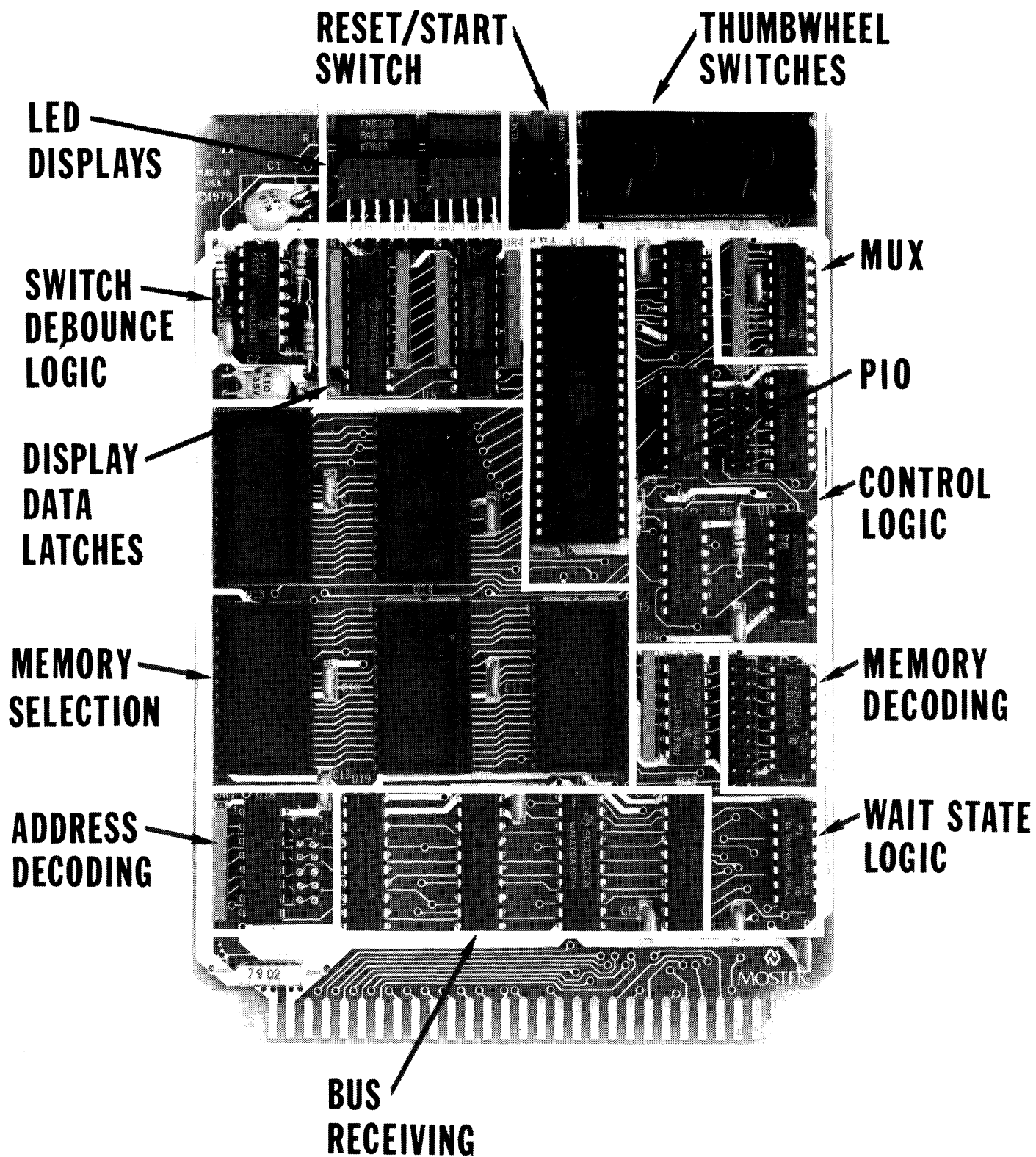
2-6. **OPERATOR INTERFACE.** The MDX-SC/D Card contains input and output devices mounted at the top of the Card which provide for the necessary user interaction with the MOSTEK provided software diagnostics package. The devices can also be utilized in a user defined application. The devices provided are:

- A. **DISPLAY** - consists of two seven segment LED displays. The display is used for indicating test results to the operator.
- B. **THUMBWHEELS** - consists of two thumbwheels each having a digit range of 0 to 9. The thumbwheels are used for operator selection of a desired diagnostic test number to be performed; possible test numbers can range from 00 to 99.

C. RESET/START SWITCH - consists of a three position toggle switch; RESET position, START position, and OFF position. The RESET and START positions are spring loaded to the OFF position. The RESET position can provide a hardware system reset. The START position can provide a begin diagnostic test function. Both the RESET and START positions have user strappable options to either the PB RESET or NMIRQ lines.

2-7. INTERRUPTS. The MK3881 PIO permits total interrupt control so that full usage of the MDX-CPU1 interrupt capabilities can be utilized during data transfers. The PIO provides vectored interrupts and maintains the daisy chain priority interrupt logic compatible with the STD-Z80 BUS.

FIGURE 2-1. BOARD PHOTO WITH OVERLAY



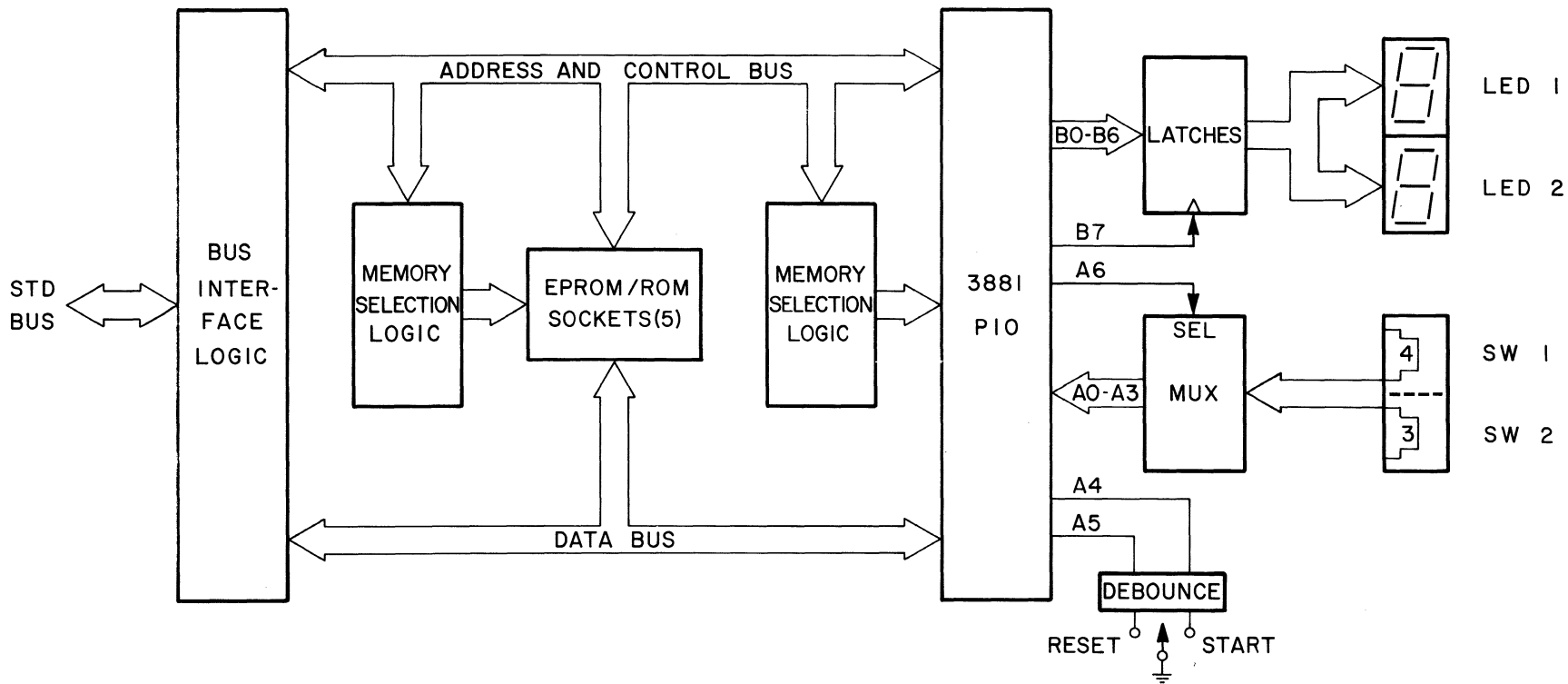


FIGURE 2-2. MDX-SC/D BLOCK DIAGRAM

SECTION 3

USER-SELECTABLE OPTIONS

3-1. EPROM DECODING JUMPER OPTIONS

3-2. The MDX-SC/D card can be populated with up to 10K x 8 of EPROM (2716's). The EPROM memories can be positioned to start on any 2K boundary within a 16K block of memory via a strapping option provided on the board. The jumper options for the decoding are shown in Table 3-1.

3-3. ADDRESS CONFIGURATION STRAPPING

3-4. The address decoding, interface, and bus management for the card are performed by the address decode and data bus circuit. The PIO ports have two addresses each and are summarized below.

	<u>PORT A</u>	<u>PORT B</u>
DATA	XX0 ₈	XX2 ₈
CONTROL	XX1 ₈	XX3 ₈

The XX symbols stand for the upper 5 bits of the I/O channel address. These bits are jumper selectable on the card in order to provide address-selectable, fully decoded ports.

3-5. RESET/START SWITCH STRAPPING

3-6. Both the RESET and START Switch positions have the following strap options.

- A. Strap to PB RESET - this can provide the user with a hardware system reset to the CPU.
- B. Strap to NMIRQ - this can provide the user with a non-maskable interrupt effect.

Regardless of whether or not a Switch is selected to be strapped, the RESET and START positions are input to PIO Port A Data for depression interpretation and/or PIO interrupt generation.

TABLE 3-1
EPROM STRAPPING OPTIONS

J3 JUMPERS						JUMPER J2
DECODED ADDRESS	EPROM U15	EPROM U14	EPROM U7	EPROM U8	EPROM U13	
0-7FF	J3 Pin to Pin 1 2	J3 Pin to Pin 3 2	J3 Pin to Pin 5 2	J3 Pin to Pin 7 2	J3 Pin to Pin 9 2	
800-FFF	" 4	" 4	" 4	" 4	" 4	Pin Pin
1000-17FF	" 6	" 6	" 6	" 6	" 6	4 to 6
1800-1FFF	" 8	" 8	" 8	" 8	" 8	and
2000-27FF	" 10	" 10	" 10	" 10	" 10	Pin Pin
2800-2FFF	" 12	" 12	" 12	" 12	" 12	2 to 3
3000-37FF	" 16	" 16	" 16	" 16	" 16	
3800-3FFF	" 14	" 14	" 14	" 14	" 14	
4000-47FF	" 2	" 2	" 2	" 2	" 2	Pin Pin
4800-4FFF	" 4	" 4	" 4	" 4	" 4	1 to 2
5000-57FF	" 6	" 6	" 6	" 6	" 6	and
5800-5FFF	" 8	" 8	" 8	" 8	" 8	Pin Pin
6000-67FF	" 10	" 10	" 10	" 10	" 10	4 to 6
6800-6FFF	" 12	" 12	" 12	" 12	" 12	
7000-77FF	" 16	" 16	" 16	" 16	" 16	
7800-7FFF	" 14	" 14	" 14	" 14	" 14	
8000-87FF	" 2	" 2	" 2	" 2	" 2	Pin Pin
8800-8FFF	" 4	" 4	" 4	" 4	" 4	2 to 3
9000-97FF	" 6	" 6	" 6	" 6	" 6	and
						Pin Pin
						4 to 5

TABLE 3-1 (CONTD)

											JUMPER J2	
DECODED ADDRESS	EPROM U15		EPROM U14		EPROM U7		EPROM U8		EPROM U13			
	J3 Pin to Pin		J3 Pin to Pin		J3 Pin to Pin		J3 Pin to Pin		J3 Pin to Pin			
800-9FFF	1	8	3	8	5	8	7	8	9	8		
000-A7FF	"	10	"	10	"	10	"	10	"	10		
800-AFFF	"	12	"	12	"	12	"	12	"	12		
000-B7FF	"	16	"	16	"	16	"	16	"	16		
800-BFFF	"	14	"	14	"	14	"	14	"	14		
000-C7FF	"	2	"	2	"	2	"	2	"	2	Pin	Pin
800-CFFF	"	4	"	4	"	4	"	4	"	4	1	to 2
000-D7FF	"	6	"	6	"	6	"	6	"	6	and	
800-DFFF	"	8	"	8	"	8	"	8	"	8	Pin	Pin
000-E7FF	"	10	"	10	"	10	"	10	"	10	4	to 5
800-EFFF	"	12	"	12	"	12	"	12	"	12		
000-F7FF	"	16	"	16	"	16	"	16	"	16		
800-FFFF	"	14	"	14	"	14	"	14	"	14		

SECTION 4

SOFTWARE FUNCTIONAL DESCRIPTION

4-1. INTRODUCTION

4-2. The software description of the MDX-SC/D Card is provided in three major subsections which outline the following:

- A. PIO Programming - details the requirements for set-up of the port configurations and interrupt control.
- B. SWITCH Strapping - details the user strap options for the RESET and START Switch and the respective software requirement to accommodate each option.
- C. MEDEX-80 - details the MOSTEK diagnostic software package.

4-3. PIO PROGRAMMING

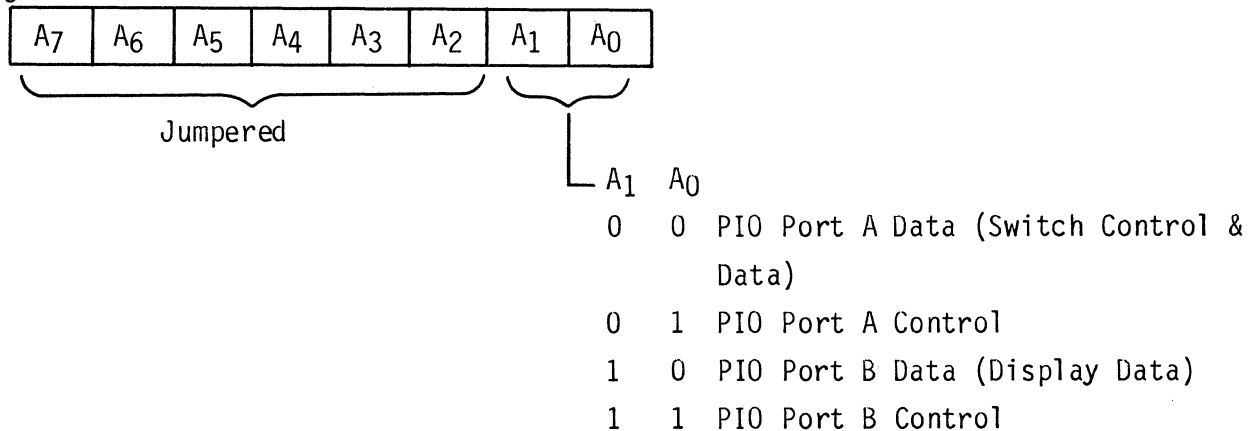
4-4. INTRODUCTION

4-5. Software interface to the MDX-SC/D Card's devices is via a Z80-PIO chip which is utilized as a pair of I/O ports for the input of thumbwheel data, the input of RESET/START Switch data, the output of display data, and to provide the interrupt circuitry of the Z-80 Mode 2 interrupt operations. The PIO must be initialized properly to ensure correct operation of the MDX-SC/D Card.

4-6. I/O ADDRESSING

4-7. The I/O addresses are jumper selectable in groups of 4 addresses anywhere within the I/O address space of the Z-80. In the I/O address depicted here, bits A₇ through A₂ are assigned by jumper selection while bits A₁ and A₀ are

assigned as indicated.



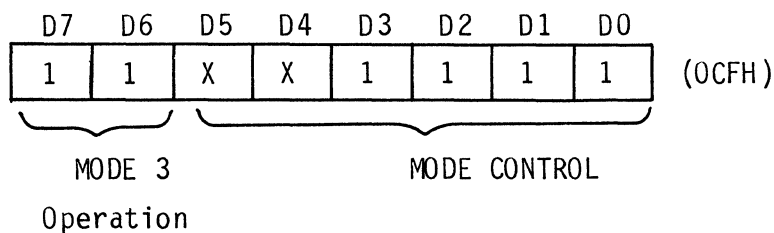
4-8. PIO OPERATION

4-9. The Z-80 PIO Port A is utilized in Mode 3 operation to provide control lines to the thumbwheels and RESET/START Switch and to provide interrupt on the condition of Switch depression. Port B is utilized in Mode 0 operation to provide an output register for the displaying of LED segment data.

4-10. PIO Port A Programming

4-11. Port A must be initialized as described in the following paragraphs. The CPU is assumed to be operating in Interrupt Mode 2; vectored daisy chain interrupt.

4-12. Mode Selection. First Port A is set to Mode 3 (control mode) operation by writing the following control word.



4-13. I/O Select. Whenever Mode 3 is selected, the next control word sent to Port A must be an I/O select word outlining the MDX-SC/D Card's I/O configuration

for Port A. A '1' indicates input line, and a '0' indicates output line. The word is:

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	1	1	1	1	1	1	(03FH)

Where:	Bit	Usage	Purpose
	D7	Output	Indicator for MEDEX INITIALIZATION complete if NMI is used.
	D6	Output	Thumbwheel selection
	D5	Input	Switches RESET position
	D4	Input	Switches START position
	D3-0	Input	Thumbwheel data

4-14. INTERRUPT VECTOR. Next the desired interrupt vector must be loaded (refer to CPU manual for details);

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

where: D0 = 0, signifies this control word is an interrupt vector.

V7 - D1 = user defined proper interrupt vector.

NOTE: This interrupt vector word is required to be loaded only if the RESET and/or START Switch is user strapped to generate a PIO vectored interrupt.

4-15. INTERRUPT CONTROL. An interrupt control word is next sent to Port A:

D7	D6	D5	D4	D3	D2	D1	D0	
1	0	1	1	0	1	1	1	(0B7H)
Enable	OR	Active	Mask					
Interrupts	Logic	High	Follows		Interrupt	Control		

NOTE: If neither the RESET nor START Switch are user strapped to generate a PIO interrupt, then the user must load 007H for the Interrupt Control Word; this will disable the PIO interrupt.

4-16. INTERRUPT MASK. The mask word following the interrupt control word selects D5 and D4 (Switch RESET position, and Switch START position) of the port data bus to produce an interrupt on user selected Switch position depression. The mask word is:

D7	D6	D5	D4	D3	D2	D1	D0
1	1	RS	SS	1	1	1	1

where: Bit Value Purpose

D5 RS RESET Switch
 0 = produce interrupt
 1 = don't produce interrupt

D4 SS START Switch
 0 = produce interrupt
 1 = don't produce interrupt

NOTE: This interrupt mask word is required to be loaded only if the RESET and/or START Switch is user configured to generate a PIO interrupt.

4-17. INTERRUPT HANDLING. Interrupts may be disabled or enabled without modifying the rest of the interrupt control word by sending the following control word to Port A Control as required:

D7	D6	D5	D4	D3	D2	D1	D0
1/0	X	X	X	0	0	1	1
Enable/Disable Interrupts	OR Logic	Active High	Mask Follows	INTERRUPT ENABLE/DISABLE CONTROL			

where: To enable interrupts, the word is 083H.
 To disable interrupts, the word is 003H.

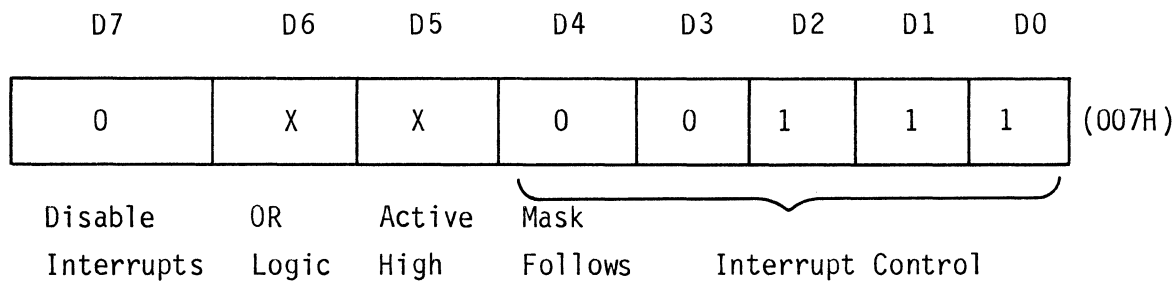
4-18. If the interrupt disable control word (003H) is used, care should be taken to disable CPU interrupts for the duration of the control word write to prevent

interference from an external asynchronous interrupt occurring during the CPU write. The code sequence to achieve this would be:

```
LD A, (003H)    ; LOAD CONTROL WORD
DI              ; DISABLE CPU INTERRUPTS
OUT (PIO), A    ; DISABLE PIO
EI              ; ENABLE CPU INTERRUPTS
```

4-19. Interrupt control need not be initialized in the enabled state, as was shown by issuing a 0B7H as the interrupt control word. Instead, a control word of 037H could be used where D7 = 0 specifies disable interrupts.

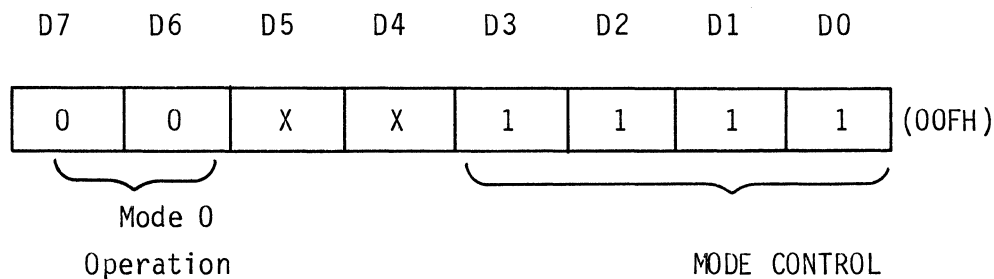
4-20. Should a system require that the interrupt feature not be utilized at all, only one interrupt control word (rather than the three above mentioned) need be sent to Port A Control. Its format follows:



4-21. PIO Port B Programming

4-22. Port B must be initialized as described in the following paragraphs.

4-23. Mode Selection. First Port B is set to Mode 0 (output mode) operation by writing the following control word.



4-24. Interrupt Control. Interrupts from PORT B must be disabled. To ensure this, the following control word is sent to Port B Control:

D7	D6	D5	D4	D3	D2	D1	D0	
0	X	X	0	0	1	1	1	(007H)

Disable OR Active Mask
 Interrupts Logic High Follows Interrupt Control

4-25. PIO Port A Data Configuration

4-26. Port A's data configuration is as follows:

1. Port A (Mode 3) Output

D7	D6	D5	D4	D3	D2	D1	D0
IC	TS	X	X	X	X	X	X

where: Bit Value Purpose

D7	IC	MEDEX Initialization Complete indicator if NMI is used. 0 = No 1 = Yes
D6	TS	Thumbwheel Select 1 = Tens Position 0 = Units Position
D5-D0	-	Used as input bits for thumbwheel data

2. Port A (Mode 3) Input

D7	D6	D5	D4	D3	D2	D1	D0
IC	X	RS	SS	TD	TD	TD	TD

where:	Bit	Value	Purpose
	D7	IC	MEDEX Initialization Complete indicator if NMI is used.
	D6	-	Used as an output bit for selecting thumbwheels
	D5	RS	RESET Switch 0 = not depressed 1 = depressed
	D4	SS	START Switch 0 = not depressed 1 = depressed
	D3-D0	TD	Thumbwheel Data*, In BCD 0000 = 0 0101 = 5 or 50 0001 = 1 or 10 0110 = 6 or 60 0010 = 2 or 20 0111 = 7 or 70 0011 = 3 or 30 1000 = 8 or 80 0100 = 4 or 40 1001 = 9 or 90

*NOTE: Thumbwheel Data (TD) read is dependent on the thumbwheel Select (TS) value, whether units or tens.

4-27. PIO Port B Data Configuration

4-28. Port B's data configuration is as follows:

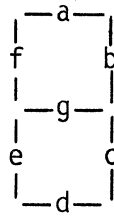
1. Port B (Mode 0) Output

D7 D6 D5 D4 D3 D2 D1 D0

LS	SD _a	SD _b	SD _c	SD _d	SD _e	SD _g	SD _f
----	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

where:	Bit	Value	Purpose
	D7	LS	Lamp Select 0 = Most Significant character Display. 1 = Least Significant character Display.
	D6-D0	SD	Segment Data for lamp selected in LS. 0 = turn on segment 1 = turn off segment

Segment labeling is



4-29. READING THE THUMBWHEELS

4-30. Reading the thumbwheel's data is accomplished by first writing a thumbwheel select word to Port A Data and then reading the BCD data from Port A data. This sequence is repeated for the other thumbwheel data value. A code sequence to input both thumbwheel's data values would be:

```

SCADAT EQU 0XXH          ; MDX-SC/D Port A Data Address.
TSUNIT EQU 000H          ; Units Thumbwheel bit select
TSTEN  EQU 040H          ; Tens Thumbwheel bit select
:
:
LD     A, TSUNIT          ; Get bit select for units thumbwheel.
OUT    (SCADAT), A        ; Set units select bit.
IN     A, (SCADAT)        ; Get BCD units value.
AND    00FH              ; Get rid of unwanted bits
LD     C,A                ; Place units in Register C
LD     A, TSTEN           ; Get bit select for tens thumbwheel.
OUT    (SCADAT)           ; Set tens select bit.
IN     A, (SCADAT), A     ; Get BCD tens value.
AND    00FH              ; Get rid of unwanted bits
LD     B,A                ; Place tens in Register B.
  
```

Register B now contains the Tens BCD value and register C contains the Units BCD value.

4-31. USING THE DISPLAY

4-32. The writing of data to the display is accomplished by writing the desired segment data and the character select position. To display a character sequence of 'F5' the code sequence would be:

```

SCBDAT EQU 0XXH      ; MDX-SC/D Port B Data Address.
CHARF  EQU 038H      ; Segments for character F.
CHAR5  EQU 024H      ; Segments for character 5.
LSLSC  EQU 080H      ; Least significant character lamp select
:
:
LD     A, CHARF      ; Get most significant character, F.
OUT    (SCBDAT), A   ; Display character F.
LD     A, CHAR5      ; Get least significant character 5.
OR     LSLSC         ; Set least significant character bit.
OUT    (SCBDAT), A   ; Display character 5.

```

The characters 'F5' now appear in the display. For applications having many different characters to display, it is recommended that a character to segment decode look-up table be used.

4-33. READING THE RESET/START SWITCH

4-34. Reading the RESET/START Switch is accomplished by reading the switch position from Port A Data. A code sequence to input the Switch position would be:

```

SCADAT EQU 0XXH      ; MDX-SC/D Port A Data Address.
SRESET EQU 5         ; RESET Switch position depressed, bit position.
SSTART EQU 4         ; START Switch position depressed, bit position.
:
IN      A, (SCADAT)   ; Get Switch position Data.
BIT     SRESET, A     ; Test if RESET Switch Side depressed.
JP      NZ, PRESET    ; If depressed, process RESET.
BIT     SSTART, A     ; Test if START Switch Side depressed.
JP      NZ, PSTART    ; If depressed, process Start.

```

: ; Neither Switch Side depressed.
:

A similar code sequence as above would also be used in an interrupt service routine if the MDX-SC/D Card's PIO Port A Control word is configured to generate a vectored interrupt on either RESET or START Switch depression.

4-35. SWITCH STRAPPING

4-36. INTRODUCTION

4-37. A user strapping option is provided for both the RESET and the START positions of the toggle switch. Both Switch positions can individually be strapped to either the STD Bus PBRESET line, or the STD Bus NMIRQ line, or to neither line. Table 4-1 outlines the user strap options for both the RESET and START Switch positions along with the MDX system reaction when the respective switch side is depressed.

4-38. CONFIGURATIONS

4-39. Strapping Configurations. As outlined in Table 4-1, the RESET/START Switch strapping option provides the user with several configuration possibilities. Care must be exercised in selecting which switch position will be strapped along with the respective correct programming set-up of the MDX-SC/D Card's Port A Interrupt Control word. Improper set-up can create a double interrupt effect when a single switch depression occurs. For example, if a switch side is strapped to the NMIRQ line and the PIO Port A is control programmed to generate a vectored interrupt, then switch depression will cause a non-maskable interrupt and a pending maskable vectored interrupt; two interrupts for one switch depression.

4-40. Recommended Strapping. If the user does not have a specific switch strapping requirement, then the recommended strapping is for the RESET Switch strap to be to the PBRESET line with no PIO interrupt, and the START Switch strap

to be to the NMIRQ line with no PIO interrupt. This configuration allows the RESET Switch to have system master reset effect, while the START Switch will have a non-maskable vectored interrupt effect. The latter effect provides a means to execute diagnostics or a user program without resetting the system and allows execution even if a maskable interrupt is hung in an active state.

To assure proper PIO interrupt control set-up which is dependent on switch strappings, Table 4-2 provides the interrupt control words required when programming PIO Port A Control of the MDX-SC/D Card.

TABLE 4-1
RESET/START SWITCH USER STRAP OPTIONS

MDX's System Reaction To

<u>Strap Option</u>	<u>RESET Switch Depression</u>	<u>Start Switch Depression</u>
No Strap	Will latch a '1' in MDX-SC/D Card's Port A Data bit D5. Will create vectored interrupt if Port is programmed to do so.	Will latch a '1' in MDX-SC/D Card's Port A Data bit D4. Will create vectored interrupt if Port is programmed to do so.
<u>Strapped to STD-BUS PBRESET line</u>	CPU control will be transferred (jump) to location 0000H or E000H, dependent on MDX-CPU Card Debug Address Strapping. All circuitry interfaced to the STD BUS RESET line will be activated RESET enable when the switch is depressed.	Same as RESET Switch depression.

Strapped to STD-BUS
NMIRQ line

CPU control will be transferred (call) to location 0066H or E066H, dependent on MDX-CPU Card Debug Address Strapping. Will latch a '1' in MDX-SC/D's Card's Port A Data Bit D5. Will create a vectored interrupt if Port is programmed to do so. Refer to paragraph 4-38 on possible double interrupt occurrence.

CPU control will be transferred (call) to location 0066H or E066H, dependent on MDX-CPU Card's Debug Address Strapping. Will latch a '1' in MDX-SC/D Card's Port A Data bit D4. Will create a vectored interrupt if Port is programmed to do so.

TABLE 4-2
PORT CONTROL LOADING REQUIREMENTS FOR SWITCH CONFIGURATIONS

SWITCH CONFIGURATION											MDX-SC/D CARD'S PIO PORT A CONTROL		
C O N F I G #	RESET SIDE					START SIDE					FOR Interrupt Vector, Load	For Interrupt Control, Load	For Interrup Mask, Load
	NO	S	N	R	W/O P P P B I I O O	NO	S	N	R	W/O P P P B I I O O			
	T	M	E			T	M	E					
	R	I	S	I	I	R	I	S	I	I			
	A	R	E	N	N	A	R	E	N	N			
	P	Q	T	T	T	P	Q	T	T	T			
1	X	-	-	X	-	X	-	-	X	-	User defined	0B7H	OCFH
2	-	X	-	-	X	X	-	-	X	-	User defined	0B7H	0EFH
3	-	-	X	-	X	X	-	-	X	-	User defined	0B7H	0EFH
4	X	-	-	X	-	-	X	-	-	X	User defined	0B7H	0DFH
5	X	-	-	X	-	-	-	X	-	X	User defined	0B7H	0DFH
6	-	X	-	-	X	-	-	X	-	X	Not Required	007H	Not Requ
7	-	-	X	-	X	-	X	-	-	X	Not Required	007H	Not Requ

Legend: X indicates option selected.

4-41. MEDEX-80

4-42. INTRODUCTION

4-43. This section discusses concepts used throughout the MEDEX-80 Software Package.

4-44. Overview. MEDEX-80 is the MOSTEK diagnostic software for the MDX series of cards. MEDEX-80 is an acronym for MDX's Expandable Diagnostic Exercise software for the Z-80. It is a modular software package designed to diagnose to the card level the faulty MDX- card(s) within a system. It will work as either a stand alone package or integrated into a user's software package. The MOSTEK package can be used in it's entirety or as selective modules applicable to a user's requirement. MEDEX-80 is designed to work with the following minimum hardware configuration.

1. MOSTEK MDX-CPU Card
2. MOSTEK MDX-SC/D Card

MEDEX-80 can reside in either RAM or EPROM. The amount of memory required is dependent on the modules selected to be included in the users system.

4-45. Features. MEDEX-80 consists of system software and diagnostics software. The system software consists of initialization, monitor, and MDX-SC/D card handler. The diagnostics software consists of specific card test programs. The diagnostic programs provided are for only those MDX cards having little or no on-card user configurability. For MDX Cards having extensive on-card user configurability, the diagnostic programs are supplied by the user if so desired. MEDEX-80 is designed to permit inclusion of user developed diagnostic programs within its' structure. The MEDEX-80 system is diagrammed in Figure 4-1. The

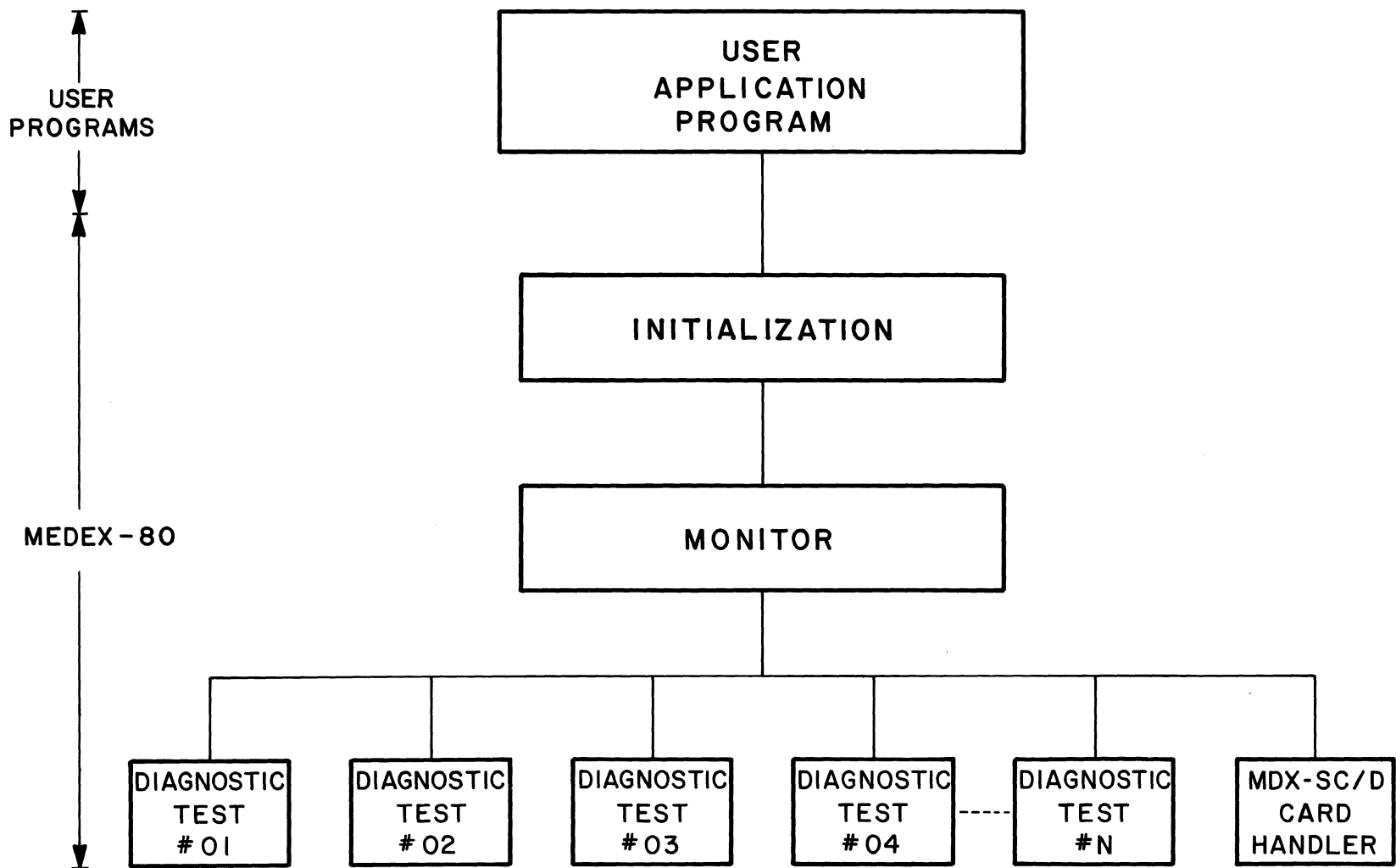


FIGURE 4-1. MEDEX-80 SYSTEM

following programs are supplied in MEDEX-80, version 1.0, and each program is discussed in detail in later sections:

1. MDINI - Initialization which provides the set-up requirements of other MEDEX-80 modules.
2. MDMON - Monitor which provides control supervision and interface to rest of diagnostic modules.
3. MDSCH - MDX-SC/D Card Handler which provides user interface to the card's I/O devices.
4. MDRAM - Diagnostic RAM test which provides confidence test of RAM areas by means of pattern tests.
5. MDROM - Diagnostic ROM test which provides confidence test of ROM and PROM areas by means of checksums.
6. MDCTC - Diagnostic CTC test which provides confidence test of MDX-CPU card's CTC by means of functional exercising.
7. MDSCD - SC/D card test which provides confidence test of MDX-SC/D Card by means of functional exercising.

4-46. THEORY OF OPERATION

4-47. The basic theory of operation of both MEDEX-80 and the MDX-SC/D Card is as follows. The user provides a CPU control transfer to the MEDEX-80 initialization module through a jump instruction or a depressed switch action. This initialization module conditions the MEDEX-80 monitor, other MDX cards that need initialization, and the MDX-SC/D Card ready for use. Once initialization is complete, the monitor takes control of the diagnostic system. To perform a test the operator selects the desired diagnostic test number via the thumbwheels. The

RESET/START Switch is then momentarily depressed to the Start position to activate the specified test number. The monitor validates the test number and transfers CPU control to the proper diagnostic test program. The display will indicate a 'test in progress' readout. If the test is permitted to execute until completion of the selected diagnostic test, then upon completion the test results will be indicated in the display. The display indication of pass or fail, and type of fail, is dependent on the diagnostic test program. A different or same diagnostic test can now be executed following the cycle as just described. If the operator desires to terminate a diagnostic test in progress, or to reinitialize the diagnostic system, the RESET/START Switch is momentarily depressed to the RESET position. The depression to the RESET position will cause the MEDEX-80 initialization module to again be executed, or cause execution of the users initialization (0000H) if Reset Switch is strapped to PBRESET.

4-48. CONCEPT OF DIAGNOSTIC MODULE

The diagnostic module is a program which executes a specific test or tests, and upon completion outputs the test results to the operator display in the form of a coded number.

Each diagnostic module is composed of the following programs, or submodules.

1. Initialization - provides the unique initialization requirements.
2. Test - provides the unique test(s) to exercise and determine specific faults and determines the pass/fail decision on performed test, and provides coded operator display.
3. ISR - provides the unique interrupt service routine handling if applicable to this diagnostic module.

4-49. CONCEPT OF TEST DIRECTIVE TABLE

4-50. The expandability ease of MEDEX-80 is accomplished through the Test Directive Table. This Table contains information on each diagnostic test within

the diagnostic system necessary for the monitor to validate and transfer CPU control.

Each diagnostic test has one (1) entry in the Test Directive Table. Each entry is structured as:

		#Of	Data		
<u>#</u>	<u>Field</u>	<u>Bytes</u>	<u>Type</u>	<u>Valid Range</u>	<u>For Contents of</u>
0	DMODT#	1	Binary	00 through 99	Diagnostic Module Test #
1	DMODIA	2	Binary	aaaa	Diagnostic Module Initialization Address
2	DMODTA	2	Binary	aaaa	Diagnostic Module Test Address

4-51. DMODT#. The diagnostic module test # field is the user specified # of this diagnostic test. This number, when operator selected in the thumbwheels, and operator depression of the RESET/START Switch to the START position, will cause the diagnostic test program located at DMODTA to be executed. The test # is a one (1) byte binary value ranging from 00 through 99.

4-52. DMODIA. The DMODIA field contains the initialization address of this diagnostic module. The initialization portion of each diagnostic module allows for each diagnostic module to initialize its own unique requirements. These requirements are generally interrupt vector address, bit port masks, interrupt control, etc. Each unique initialization portion generally conditions its unique hardware to a quiescent state. Every time MEDEX-80 is entered from a user program and every time the RESET/START Switch is depressed to the RESET position (dependent on strap option), the monitor will give CPU control to each diagnostic module, one at a time, at the diagnostic test program's initialization program located at DMODIA. The DMODIA is a two (2) byte binary value which represents the address of the initialization program. When no initialization of the module is required, the program at this DMODIA location will be only an RET instruction.

4-53. DMODTA. The DMODTA field contains the test address of this diagnostic mod-

ule. When the thumbwheel entry equals the test # field, and the RESET/START Switch is depressed to the START position, the monitor will transfer CPU control to the specified address in this field. The DMODTA is a two (2) byte binary value which represents the address of the diagnostic test module.

4-54. Entry Example. For a diagnostic module of CRT test with the following; a test # of 15, an initialization entry point of DCRTI at location 2731_H, and a diagnostic module entry point of DCRTT at location 2750_H, the Test Directive Table entry would look like:

<u>For Source Code</u>				<u>For Assembled Linked Object Code</u>
T15	DEFB	15	;Test # 15	0F
	DEFW	DCRTI	;CRT diagnostic initiali- zation	3127
	DEFW	DCRTT	;CRT diagnostic test	5027

4-55. End Of Table Indicator. The last entry in the Test Directive Table is an end of Table indicator. This indicator is the identical structure as each diagnostic test entry except that only the DMODT# field is required. The DMODT# field contains FFH for end of Table indication.

4-56. Start Of Table Label. The first diagnostic module entry in the Test Directive Table must be either labeled or equated to label M?TDS. The monitor uses this label as a base address for its process requirements.

4-57. Table Examples. Figure 4-2 illustrates a source code example of the Test Directive Table. In this example all diagnostic test entries are ordered sequentially, although non-sequential entries are permitted also.

4-58. Expansion. To add another diagnostic test module to MEDEX-80, simply create the required Test Directive Table entry and insert between the last entry and the end of table indicator. Reassemble the Table with the new diagnostic

modules. The additional diagnostic modules are now structured into MEDEX-80.

4-59. Relationship Of Diagnostic Module To Test Directive Table. Figure 4-3 illustrates the relationship of a diagnostic module to the Test Directive Table. In all cases, the Test Directive Table entry fields of DMODIA and DMODTA are the entry points into the respective diagnostic modules of Initialization and Test.

4-60. CONCEPT OF DIAGNOSTIC INTERRUPT VECTOR TABLE

4-61. The Diagnostic Interrupt Vector Table provides the capability to test, if so desired, the corresponding interrupt feature if applicable within a diagnostic test module. This Table contains information on each diagnostic vector necessary for proper transfer of CPU control during interrupt occurrences. MEDEX-80 operates under Z-80/Mode 2 Interrupt.

Each testable interrupt for each diagnostic module has one (1) entry in the Diagnostic Interrupt Vector Table. Each entry is structured as:

# Of				
<u>Field</u>	<u>Bytes</u>	<u>Data Type</u>	<u>Valid Range</u>	<u>For Contents of</u>
DMODSA	2	Binary	aaaa	Diagnostic Module ISR Address

4-62. DMODSA. The DMODSA field contains the address of the diagnostic module's interrupt service routine. This routine will handle any interrupt occurrences from its respective MDX card during any process within MEDEX-80. This allows each diagnostic module to process its own unique interrupt requirements. The DMODSA field is a two (2) byte binary value which represents the address of the module's interrupt service routine. Diagnostic modules not utilizing interrupts as part of their test do not have to provide a DMODSA entry.

4-63. Entry Example. For a diagnostic module of CRT Test with the following; an interrupt service routine entry point of DCRTIS at location 2825H, with a table

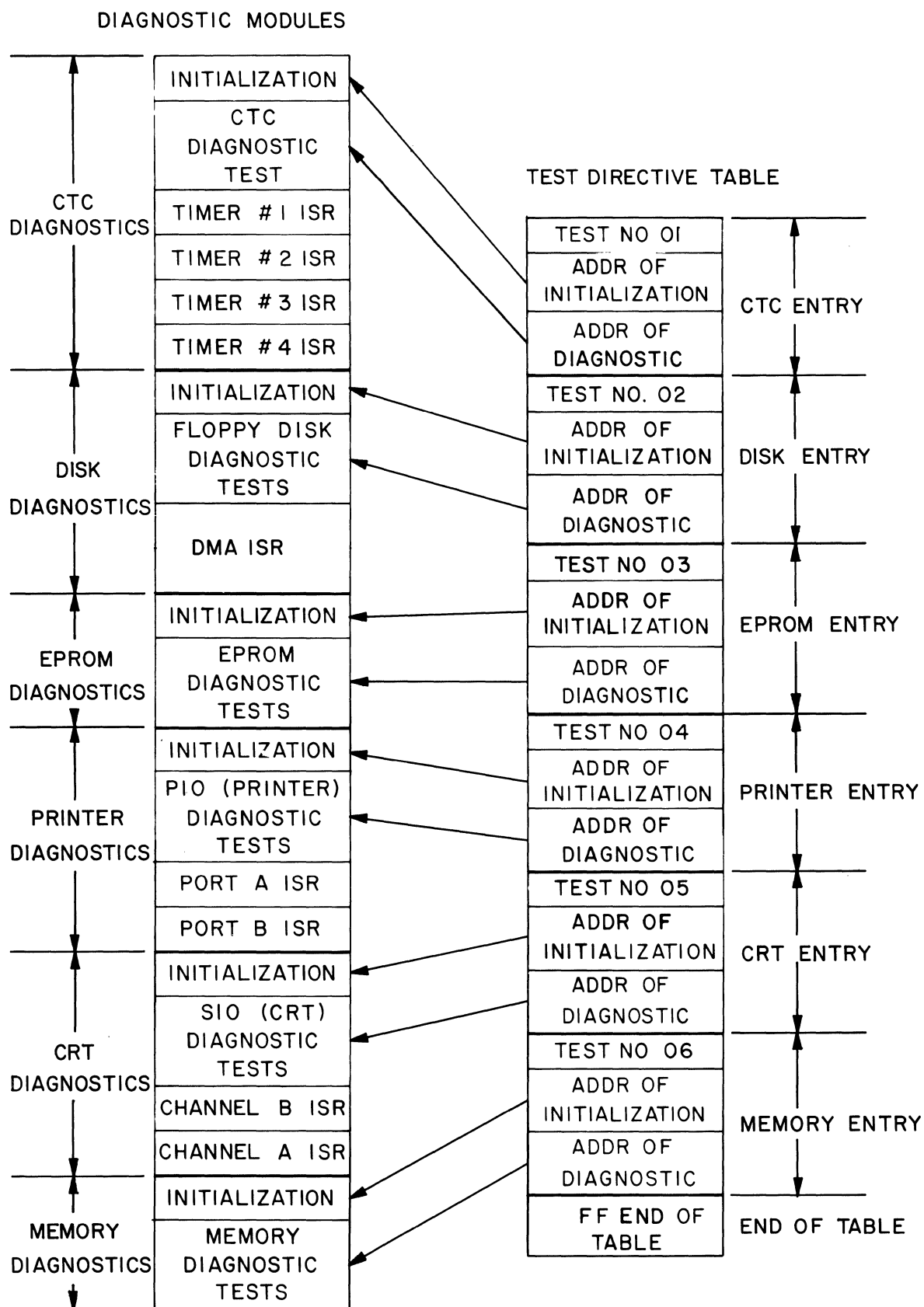
FIGURE 4-2. TEST DIRECTIVE TABLE SOURCE CODE EXAMPLE

```

TITLE    MEDEX-B0 TEST DIR TABLE EXAMPLE
;
;
;
;
;      MEDEX - B0
;      TEST DIRECTIVE TABLE
;
;
;      EACH DIAGNOSTIC MODULE ENTRY IS STRUCTURED AS;
;
;      FIELD #      FIELD      CONTENTS
;      -----
;      0            TEST#      TEST # OF DIAGNOSTIC MODULE
;      1            INITA      INITIALIZATION ADDRESS OF DIAGNOSTIC MODULE
;      2            DIAGA      DIAGNOSTIC TEST ADDRESS OF DIAGNOSTIC MODULE
;
;
;      GLOBAL M?CTCI
;      GLOBAL M?CTCT
;      GLOBAL M?RAMT
;      GLOBAL M?RAMI
;      GLOBAL M?ROMT
;      GLOBAL M?ROMI
;      GLOBAL M?SCDI
;      GLOBAL M?SCDT
;      GLOBAL PRINTI
;      GLOBAL PRINTT
;
;
;
M?TDS EQU $           ; START OF TEST DIRECTIVE TABLE
T01  DEFB 01          ; TEST 01, CTC TEST
      DEFW M?CTCI      ; MDCTC INITIALIZATION
      DEFW M?CTCT      ; MDCTC TEST
T02  DEFB 02          ; TEST 02, RAM TEST
      DEFW M?RAMI      ; MDRAM INITIALIZATION, NO INIT REQUIRED
      DEFW M?RAMT      ; MDRAM TEST
T03  DEFB 03          ; TEST 03, ROM TEST
      DEFW M?ROMT      ; MDROM INITIALIZATION, NO INIT REQUIRED
      DEFW M?ROMT      ; MDROM TEST
T04  DEFB 04          ; TEST 04, MDX-MATH CARD TEST
      DEFW M?SCDI      ; MDSCD INITIALIZATION
      DEFW M?SCDT      ; MDSCD TEST
T05  DEFB 05          ; TEST 05, PRINTER TEST
      DEFW PRINTI      ; PRINTER DIAGNOSTIC INITIALIZATION
      DEFW PRINTT      ; PRINTER DIAGNOSTIC TEST
M?TDE DEFB 0FFH       ; END OF TABLE
      END

```

FIGURE 4-3. DIAGNOSTIC MODULE RELATIONSHIP TO TEST DIRECTIVE TABLE



entry reference label of DCISR, the Diagnostic Interrupt Vector Table entry would look like:

For Source Code

For Assembled Linked Object Code

DCISR DEFW DCRTIS ;CRT diagnostic ISR	2528
---------------------------------------	------

DCISR is the Address to be used to write to its respective hardware interrupt Vector; least significant 8 bits.

4-64. Start Of Table Label. The first diagnostic module entry in the Diagnostic Interrupt Vector Table must be either labeled or equated to label M?DITS. The MEDEX-80 initialization process uses the upper eight (8) bits of this label's address for the value to be loaded in the I register.

4-65. Table Example. Figure 4-4 illustrates a source code example of the Diagnostic Interrupt Vector Table.

4-66. Relationship Of Diagnostic Module To Diagnostic Interrupt Vector Table. Figure 4-5 diagrams the relationship of a diagnostic module to the Diagnostic Interrupt Vector Table. The order of entries in the Table are not required to be in any specified sequence.

4-67. INTERRUPTS

4-68. Interrupt Vector Operation. Each user desired to-be-tested interrupt capability of each MDX card is initialized during the MEDEX-80 initialization process. Each diagnostic module's initialization process will load its respective interrupt vector(s) on its respective MDX card. The value used for the vector address is the lower eight (8) bits of the label value for the modules's entry in the Diagnostic Interrupt Vector Table. MEDEX-80 will initialize the CPU's I register during its initialization. Whenever an interrupt occurs, the MDX Card's interrupt vector is linked with the I register forming an address which points to the respective entry address of the Diagnostic Interrupt Vector Table. An

indirect branch to the address within this entry is then performed by the CPU and process control is transferred to the specified interrupt service routine.

4-69. Interrupt Handling. The handling of an interrupt from an MDX Card is the responsibility of the respective Card's diagnostic module. If a user elects not to involve interrupts within the diagnostic module test, then during the diagnostic module's initialization, the user should configure the MDX Card's interrupt control word to a disable interrupt state. During MEDEX-80 initialization phase, the Z-80 system wide interrupt capability is disabled. MEDEX-80 will enable system wide interrupt capability after all diagnostic module have completed their unique initialization. During each diagnostic modules initialization, the user should disable the unique MDX Card's interrupt. This will prevent unsolicited interrupts from occurring. When the diagnostic module is selected to execute, the first operation should be to enable its respective MDX Card's interrupt capability, and upon completion of the test, the Card's interrupt capability should be disabled.

4-70. Unsolicited Interrupt Handling. If a user elects to allow interrupts to be always enabled on any card regardless of whether or not the respective diagnostic module is in execution, the user should provide for unsolicited interrupt handling. This handling should accommodate interrupt processing for interrupt occurrences when the respective diagnostic module is not currently active, and when a different card's interrupt erroneously vectors to this diagnostic module. The latter condition can happen when another diagnostic module is active and its card's interrupt vector logic is in fault. MEDEX-80 provides an indication as to who the owner of this stray interrupt may belong to. Whenever a diagnostic module is active, MEDEX-80 places the address of the active diagnostic module in register IY. The user should check in the diagnostic module's interrupt service routine the IY register contents for a match to its own diagnostic address. If no match exists and the inactive diagnostic module's MDX Card's interrupt was not the instigating device, then the interrupt belongs to another diagnostic module. When this occurs the user should transfer CPU control to MEDEX-80 unsolicited interrupt handler (M?UNSI) for proper disposition; later sections will provide further information.

FIGURE 4-4. DIAGNOSTIC INTERRUPT VECTOR TABLE SOURCE CODE EXAMPLE

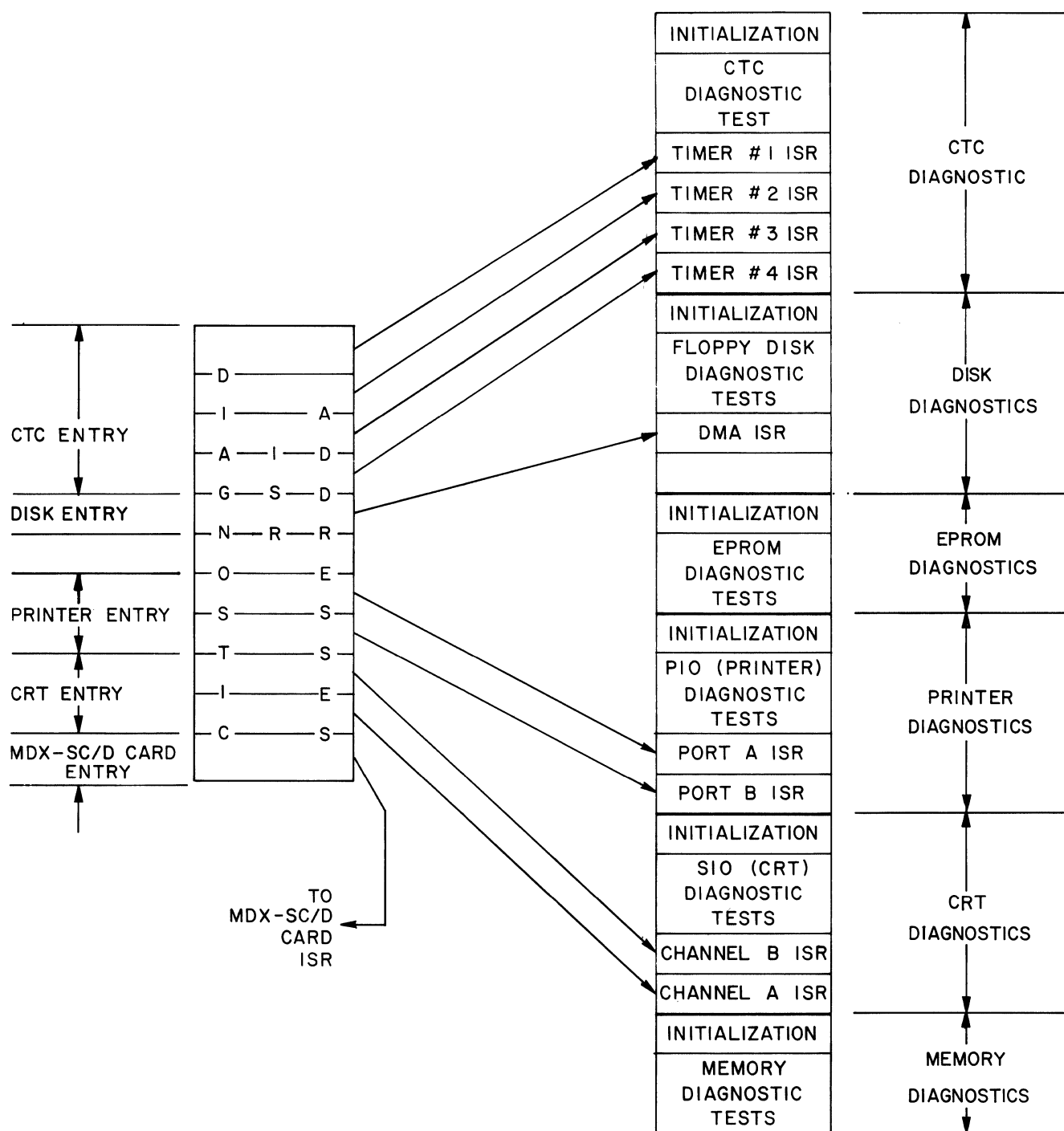
```

TITLE    MEDEX-80 DIV TABLE EXAMPLE

$
$
$
$
$      MEDEX - 80
$      DIAGNOSTIC INTERRUPT VECTOR TABLE
$
$      EACH DIAGNOSTIC TEST ENTRY IS STRUCTURED AS:
$
$      FIELD #      FIELD      CONTENTS
$      -----      -
$      0            DIAGIA     DIAGNOSTIC MODULE'S ISR ADDRESS
$
$
$
$
$      GLOBAL  M?DITS
$      GLOBAL  M?CTCV
$      GLOBAL  M?CTS0
$      GLOBAL  M?CTS1
$      GLOBAL  M?CTS2
$      GLOBAL  M?CTS3
$      GLOBAL  M?SCDS
$      GLOBAL  M?SCDV
$      GLOBAL  DIVPTR
$      GLOBAL  PRINTS
$
$
$      ORG      ($-7).AND.0FFF8H      ; FORCES START LOCATION OF 0XXX0H OR 0XX
M?DITS EQU      $                    ; START OF DIAG INTERRUPT VECTOR TABLE
M?CTCV DEFW     M?CTS0                ; ISR FOR CTC TIMER 0
      DEFW     M?CTS1                ; ISR FOR CTC TIMER 1
      DEFW     M?CTS2                ; ISR FOR CTC TIMER 2
      DEFW     M?CTS3                ; ISR FOR CTC TIMER 3
M?SCDV DEFW     M?SCDS                ; ISR FOR MDSCD DIAGNOSTICS
DIVPTR DEFW     PRINTS                ; ISR FOR PRINTER DIAGNOSTICS
      END

```


FIGURE 4-5. DIAGNOSTIC MODULE RELATIONSHIP TO DIAGNOSTIC INTERRUPT VECTOR TABLE



4-71. DIAGNOSTIC SYSTEM ARCHITECTURE

4-72. Architecture. The diagnostic architecture of a constructed MEDEXS-80 package is diagrammed in Figure 4-6. This figure diagrams the diagnostic modules and their relationship to both the Test Directive Table and the Diagnostic Interrupt Vector Table and includes the monitor, initialization, and MDX-SC/D Card handler.

4-73. System Entry Point. The entry point for starting MEDEX-80 in either a stand alone or an integrated configuration is MEDEX.

4-74. Stand Alone Configuration. If the package is to be used totally by itself, then MEDEX-80 should be origin assembled starting at location 0000H, or E000H, dependent on the user's MDX-CPU Card's strapping of the starting address.

4-75. MEDEX-80 DISKETTE

4-76. A floppy diskette formatted in IBM 3740 soft sector single density fashion is provided with this Card. The diskette is readable on a FLP-80DOS/MDX System or an AID-80F System. The diskette includes the following.

4-77. Modules. An assembly source code file of each of the MEDEX-80 modules is provided on the diskette.

4-78. User Manuals. A software user manual for each of the above source modules is provided on the diskette. The user manual files are in ASCII text format for output to a printer. Each user manual has a unique section number which permits, once all sections are printed, the assembling into one MEDEX-80 User Manual. Appendix D illustrates the MEDEX-80 operator display characters produced by the handler.

4-79. Medex Directory. A file is provided which contains a directory of all MEDEX-80 modules and user manuals provided on the diskette. The directory

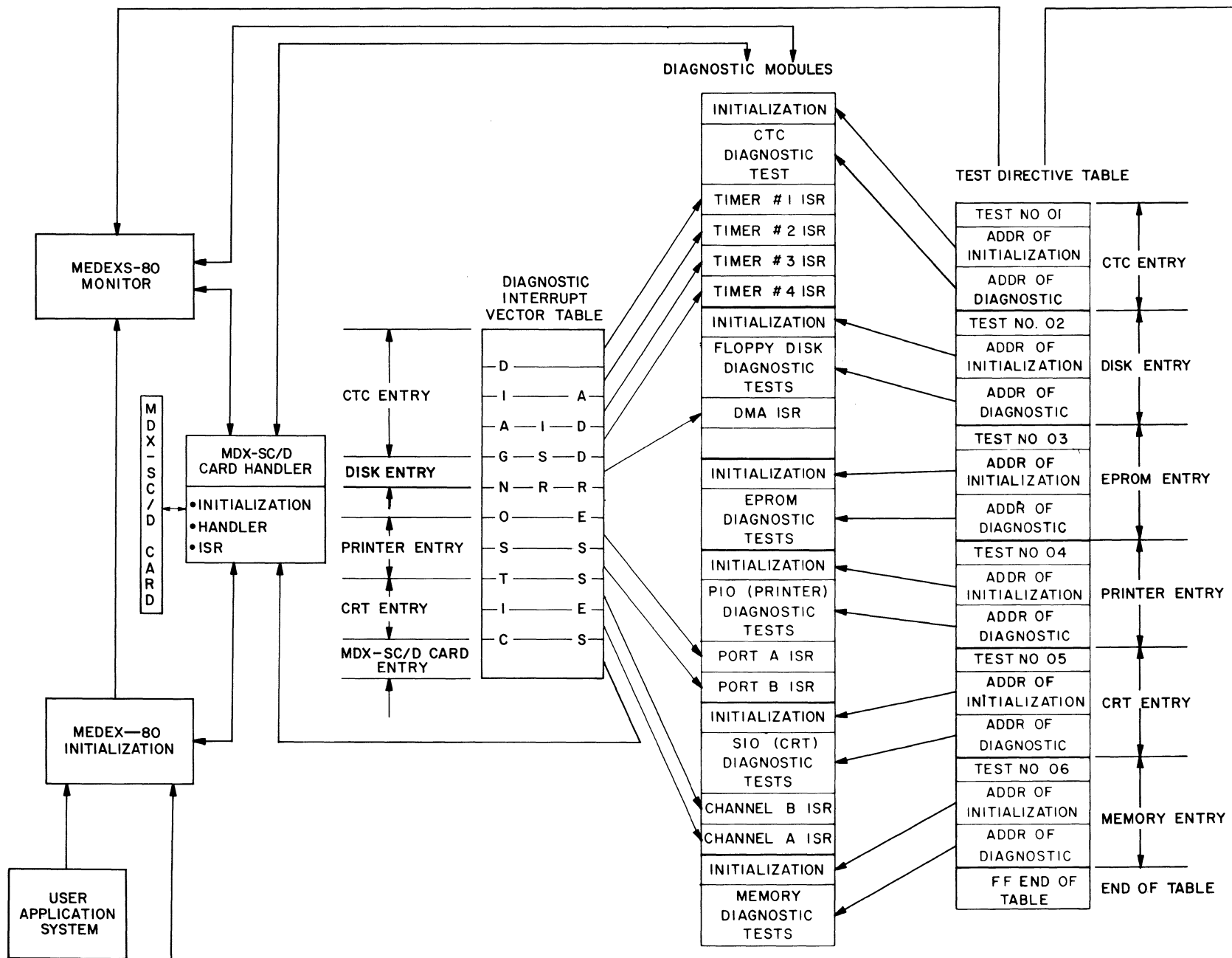


FIGURE 4-6. MEDEX-80 SYSTEM ARCHITECTURE

information includes filename, version, and description of file's contents. The directory filename is MEDEX.DOC and is in ASCII text format for output to a printer.

4-80. Labelling Conventions. All MOSTEK supplied MEDEX-80 modules adhere to a labelling convention to minimize labelling conflicts with user programs. All MOSTEK modules have their user call and reference labels begin with the characters 'M?'.

4-81. System Construction. To construct a MEDEX-80 System, the user is required to perform the following. All the parameters required to be user specified are included in a configuration file (MDCON.SRC). The user must edit this file to specify the parameters of the modules the user intends to include in the System, and delete those parameters of the modules not intended to be used in the system. Once this file is edited, the MEDEX-80 System can be constructed by assembling the edited MDCON.SRC file with all other user selected MEDEX-80 modules and user created modules. A MEDEX-80 System tailored to the user application will then exist. It is recommended that each module be individually assembled and then linked into the desired system configuration. Additional information on system construction is provided in a user manual file included on the diskette.

4-82. SUMMARY

Diagnostic System Data Structures summary is as follows.

A. For Test Directive Table (DSTD):

1. Start of Table Label - M?TDS.
2. Each Diagnostic Module Entry Requires,
 - TEST # - 1 byte binary; user specified test #
 - DMODIA - 2 byte binary, initialization address.
 - DMODTA - 2 byte binary, test address.
3. End of Table Indicator - OFFH in TEST # field.

B. For Diagnostic Interrupt Vector Table (DIVT):

1. Start of Table Label - M?DITS
2. Each Diagnostic Module Interrupt Entry Requires,
 - DMODSA - 2 byte binary, ISR address.

SECTION 5

SPECIFICATIONS

5-1. INTRODUCTION

5-2. This section describes all electrical and mechanical specifications for the MDX-SC/D board.

5-3. ELECTRICAL SPECIFICATIONS

5-4. WORD SIZE

DATA: 8 bits

I/O ADDRESSING: 8 bits

5-5. MEMORY AND I/O ADDRESSING

MEMORY: 2K blocks jumper selectable for any 2K boundary within a given 16K boundary of Z80 memory map.

I/O: On board fully programmable.

5-6. MEMORY CAPACITY

10K Bytes of 2716 memory

(2716's not included)

5-7. MEMORY SPEED REQUIRED

MEMORY	ACCESS TIME	CYCLE TIME
* 2716	450 nsec	450 nsec

* Single 5 volt type required

5-8. INTERRUPTS

Vectored interrupts generated. Interrupt vector programmable upon initialization.

Daisy chained interrupt priority. Selected bit channels can be masked out under program control.

5-9. SYSTEM CLOCK

	<u>MIN</u>	<u>MAX</u>
MDX-SC/D	250 KHz	2.5 MHz
MDX-SC/D-4	250 KHz	4.0 MHz

5-10. PARALLEL BUS INTERFACE-STD-Z80 BUS COMPATIBLE

INPUTS: One 74LS Load Max

BUS OUTPUTS: I_{OH} = -15mA min at 2.4 volts

I_{OL} = 24ma min at 0.5 volts

5-11. POWER SUPPLY REQUIREMENTS

+5 volts $\pm$ 5% at 1.2A max

5-12. OPERATING TEMPERATURE RANGE

0°C to 50°C

5-13. MECHANICAL SPECIFICATIONS

5-14. CARD DIMENSIONS

4.5 in. (11.43cm) high by 6.50 in. (16.51cm) long

0.48 in. (1.22cm) maximum profile thickness

0.062 in. (0.16cm) printed circuit board thickness

5-15. CONNECTORS

FUNCTION	CONFIGURATION	MATING CONNECTOR
STD-Z80 BUS	56 pin dual 0.125 in. centers	Printed Circuit Viking 3VH28/1CE5 Wire Wrap Viking 3VH28/1CND5
		Solder Lug Viking 3VH28/1CN5

APPENDIX A
FACTORY REPAIR
LIMITED WARRANTY

FACTORY REPAIR SERVICE

In the event that difficulty is encountered with this unit, it may be returned directly to MOSTEK for repair. This service will be provided free of charge if the unit is returned within the warranty period. However, units which have been modified or abused in any way will not be accepted for service, or will be repaired at the owner's expense. When returning a circuit board, place it inside the conductive plastic bag in which it was delivered to protect the MOS devices from electrostatic discharge. THE CIRCUIT BOARD MUST NEVER BE PLACED IN CONTACT WITH SYTROFOAM MATERIAL. Enclose a letter containing the following information with the returned circuit board.

Name, address, and phone number of purchaser

Data and place of purchase

Brief description of the difficulty

Mail a copy of this letter SEPARATELY to:

in USA:

MOSTEK Corporation
Microcomputer Service Manager
1215 West Crosby Road
Carrollton TX, 75006

OUTSIDE USA:

Please address the letter and board
to the Mostek office or represent-
tive in your country.

Securely package and mail the circuit board, prepaid and insured, to the same address.

LIMITED WARRANTY

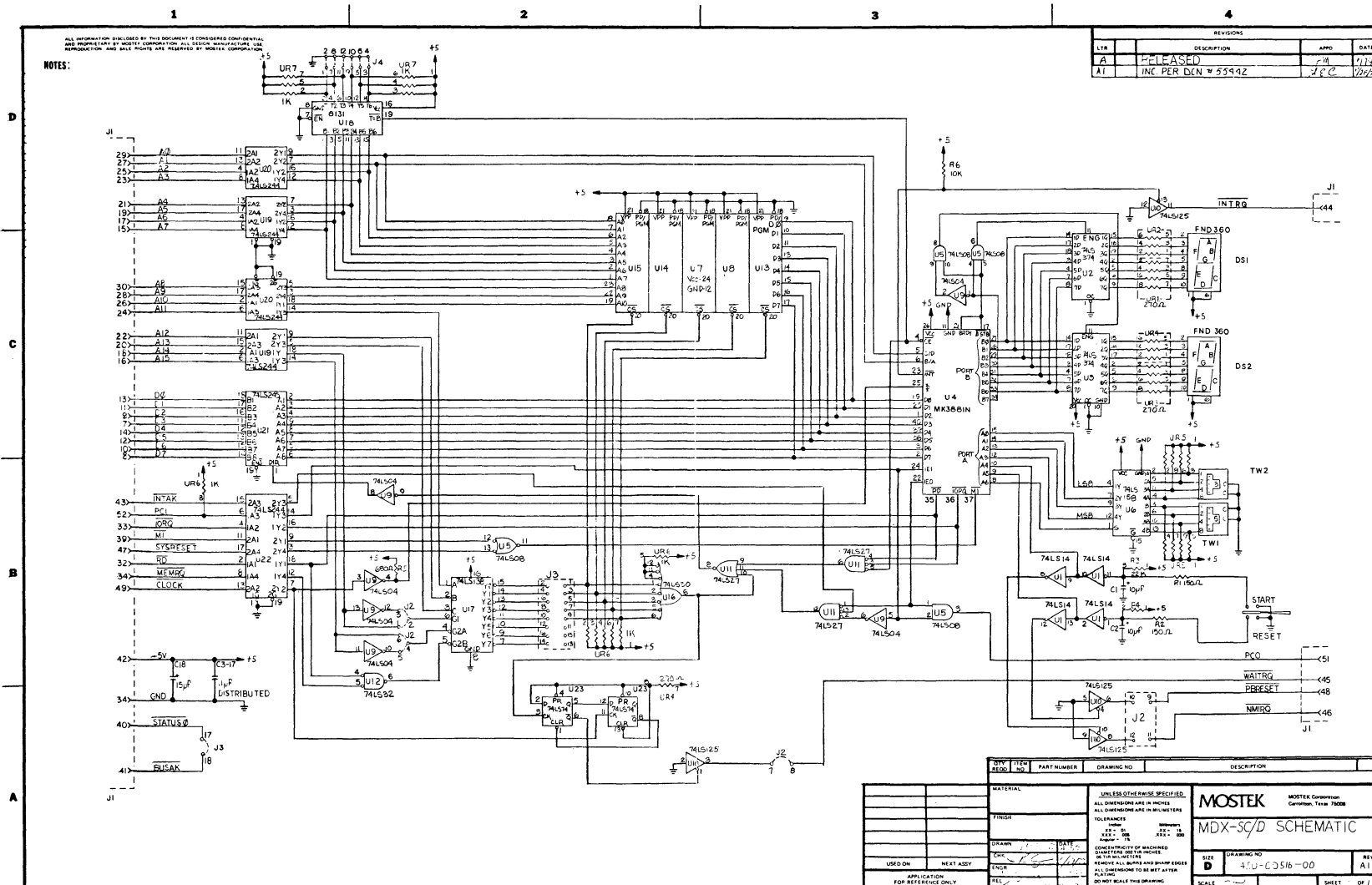
MOSTEK warrants this product against defective materials and workmanship for a period of 90 days. This warranty does not apply to any product that has been subjected to misuse, accident, improper installation, improper application, or improper operation, nor does it apply to any product that has been repaired or altered by other than an authorized factory representative.

There are no warranties which extend beyond those herein specifically given.

APPENDIX B

SCHEMATIC

SCHEMATIC

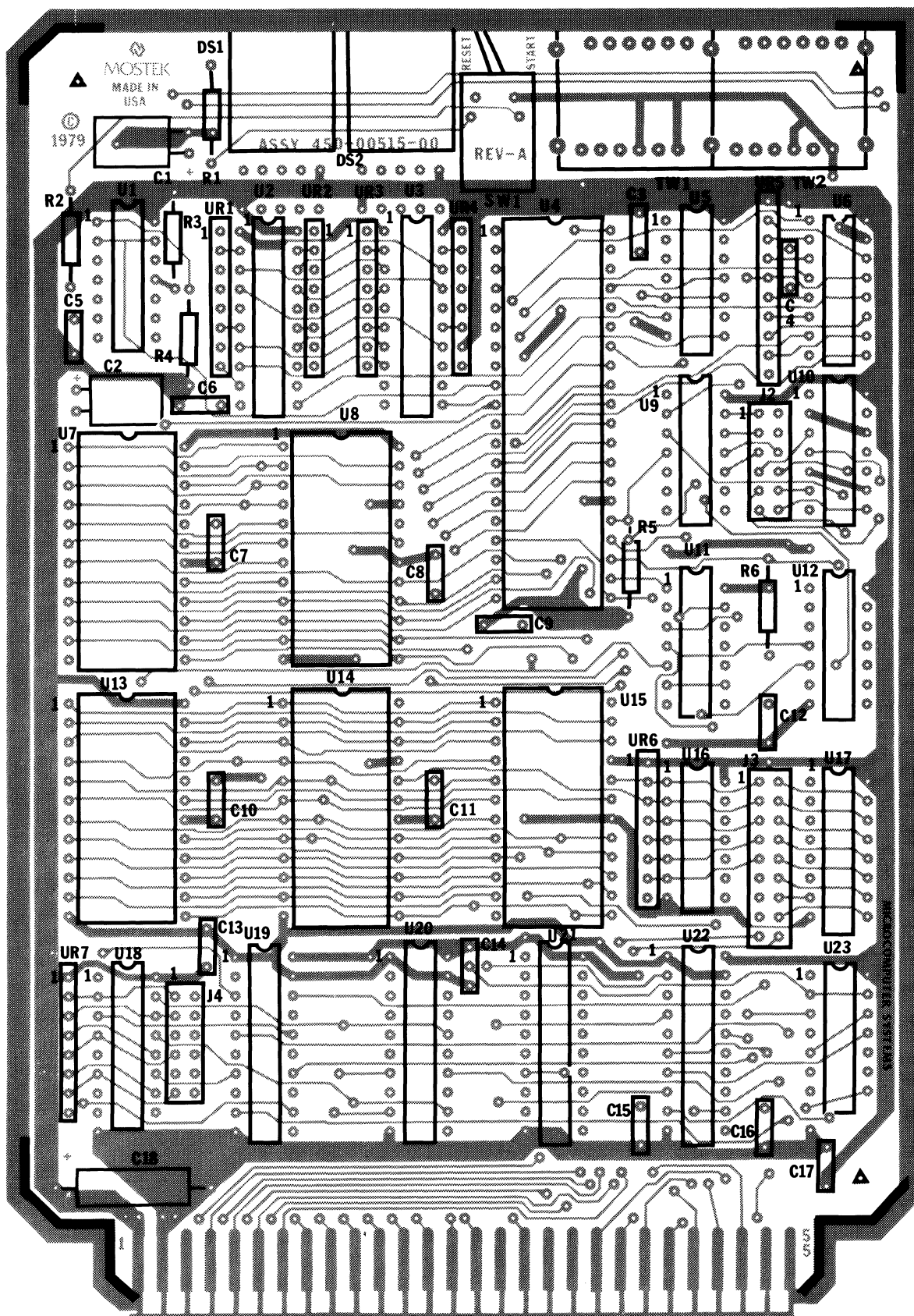


APPENDIX C

PARTS PLACEMENT DIAGRAM

PARTS LIST

ASSEMBLY DRAWING



[illegible]

APPENDIX D

MEDEX-80 OPERATOR DISPLAY CHARACTERS

APPENDIX D

MEDEX-80 OPERATOR DISPLAY CHARACTERS

Figure D-1 illustrates the operator display characters output by the MDX-SC/D Card software handler. Directly below each 7 segment character is the corresponding ASCII character and its hexadecimal value.

The handler's valid ASCII character range is 20H to 5AH. For ASCII characters outside this range, the handler will display a ' '.

SP(20H)	!(21H)	"(22H)	#(23H)	\$(24H)	%(25H)	&(26H)	'(27H)	((28H)	)(29H)	*(2AH)	+(2BH)
,(2CH)	-(2DH)	.(2EH)	/(2FH)	0(30H)	1(31H)	2(32H)	3(33H)	4(34H)	5(35H)	6(36H)	7(37H)
8(38H)	9(39H)	:(3AH)	;(3BH)	<(3CH)	=(3DH)	>(3EH)	?(3FH)	@(40H)	A(41H)	B(42H)	C(43H)
D(44H)	E(45H)	F(46H)	G(47H)	H(48H)	I(49H)	J(4AH)	K(4BH)	L(4CH)	M(4DH)	N(4EH)	O(4FH)
P(50H)	Q(51H)	R(52H)	S(53H)	T(54H)	U(55H)	V(56V)	W(57H)	X(58H)	Y(59H)	Z(5AH)	

NOTE: Characters with no segments darkened represent "blank" display output codes (20H)

FIGURE D-1 MEDEX-80 OPERATOR DISPLAY CHARACTERS

MOSTEK®
Z80-F8 Covering the full
spectrum of
3870 microcomputer
applications.

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/323-6000
In Europe, Contact: **MOSTEK Brussels**
150 Chaussee de la Hulpe, B1170, Belgium;
Telephone: 660.69.24

Mostek reserves the right to make changes in specifications at any time and without notice. The information furnished by Mostek in this publication is believed to be accurate and reliable. However, no responsibility is assumed by Mostek for its use; nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Mostek.

PRINTED IN USA October 1979
Publication No. MK79678

Copyright 1979 by Mostek Corporation
All rights reserved